

# Anhang A

## Farbkonstanz: Evaluierungen

Dieses Kapitel präsentiert einen Auszug aus den Testreihen zur Evaluierung des in Kapitel 3.2.4 vorgestellten parametrierbaren *Gray-World* Algorithmus. Zu jedem Testbild wurde ein Test-Set bestehend aus zehn farbmanipulierten Derivaten erstellt (siehe Abbildung A.1). Tabelle A.1 zeigt die entsprechenden pro Kanal verwendeten Manipulationsfaktoren. Zur Bestimmung eines Fehlervektors wurden für alle Farbkanäle die Quotienten aus

Tabelle A.1: Parameter der Testreihen

Bild Nr.	Modifikation		
	Rot	Grün	Blau
1	1,3	1,0	1,0
2	1,0	1,3	1,0
3	1,0	1,0	1,3
4	1,1	1,3	1,0
5	0,9	0,8	1,1
6	1,1	1,0	1,3
7	1,1	0,9	1,0
8	1,1	1,15	0,95
9	1,0	1,0	0,8
10	0,8	1,0	1,1

den durchschnittlichen Farbwerten des Originalbildes und der korrigierten Derivate gebildet. Ein adäquates Maß für den resultierenden Farbfehler ist die maximale Differenz zweier Vektorkomponenten. Dargestellt wurde ausschließlich der Wertebereich zwischen 0 und 1, wobei kleinere Werte auf eine höhere Güte des Algorithmus schließen lassen. Zudem wird in einer weiteren Grafik die jeweilige Anzahl selektierter Pixel dargestellt.



Bild A.1: Beispiel eines Test-Sets



(a) Gray-World Original



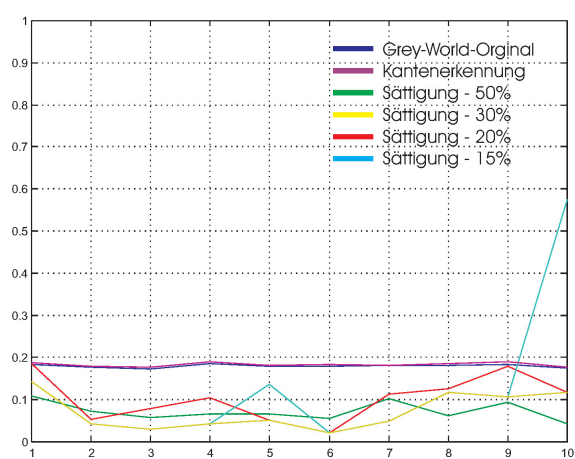
(b) mit Kantenerkennung



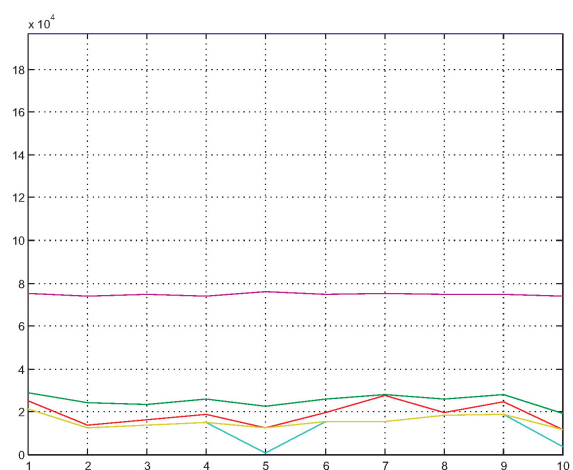
(c) mit Sättigungsgrenze bei 50%



(d) mit Sättigungsgrenze bei 20%

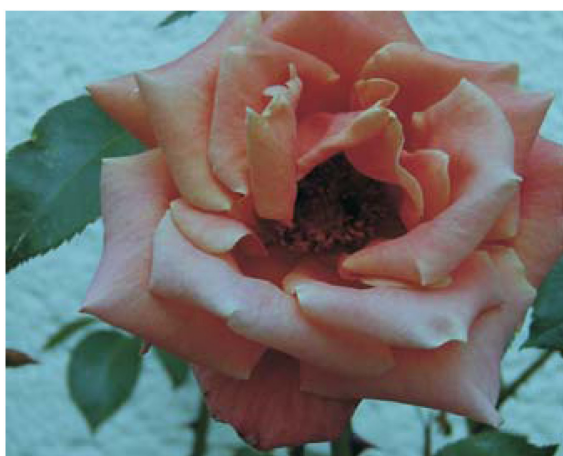


(e) Abweichung des Korrekturvektors



(f) Anzahl berechneter Pixel

Bild A.2: Gray-World Fallstudie 1 [129]



(a) Gray-World Original



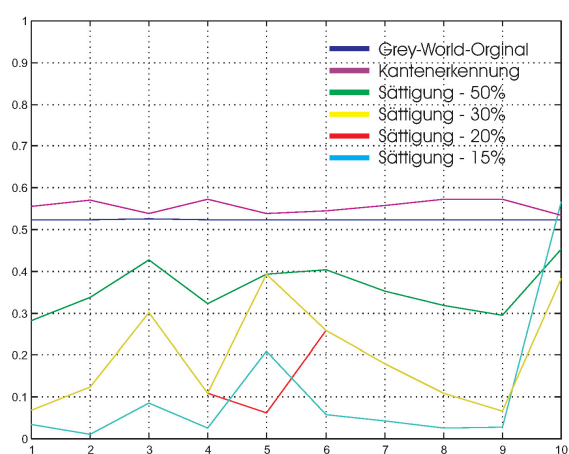
(b) mit Kantenerkennung



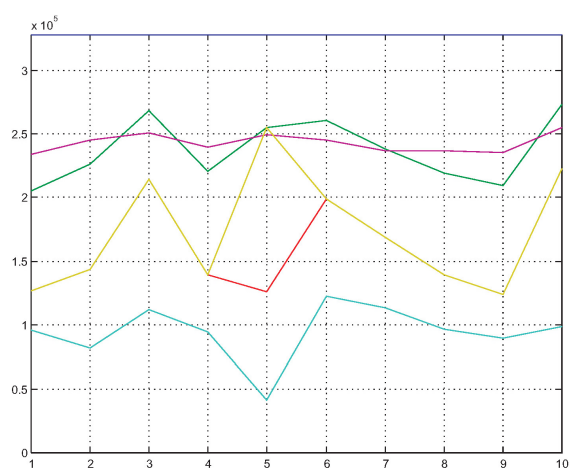
(c) mit Sättigungsgrenze bei 50%



(d) mit Sättigungsgrenze bei 20%



(e) Abweichung des Korrekturvektors



(f) Anzahl berechneter Pixel

Bild A.3: Gray-World Fallstudie 2 [129]





(a) Gray-World Original



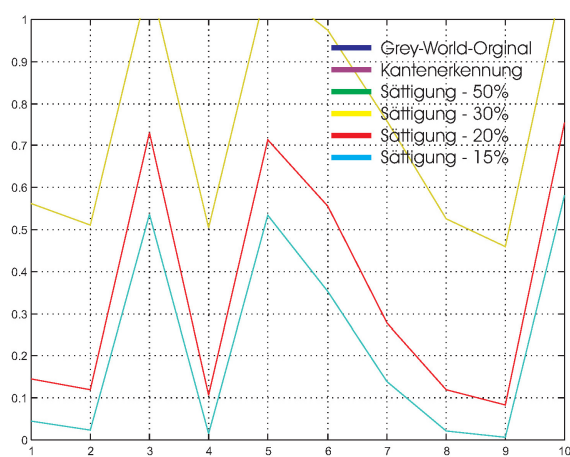
(b) mit Kantenerkennung



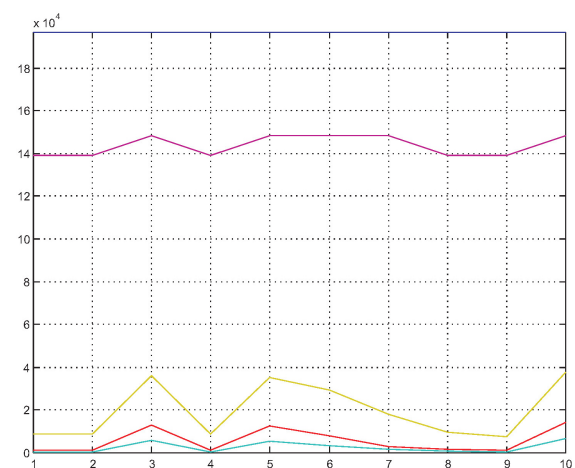
(c) mit Sättigungsgrenze bei 50%



(d) mit Sättigungsgrenze bei 20%



(e) Abweichung des Korrekturvektors



(f) Anzahl berechneter Pixel

Bild A.4: Gray-World Fallstudie 3 [129]





(a) Gray-World Original



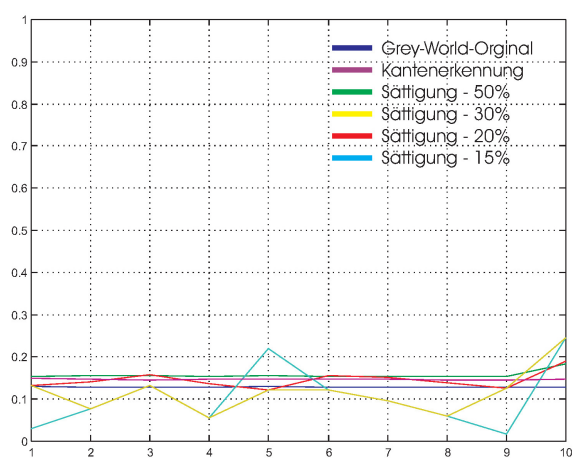
(b) mit Kantenerkennung



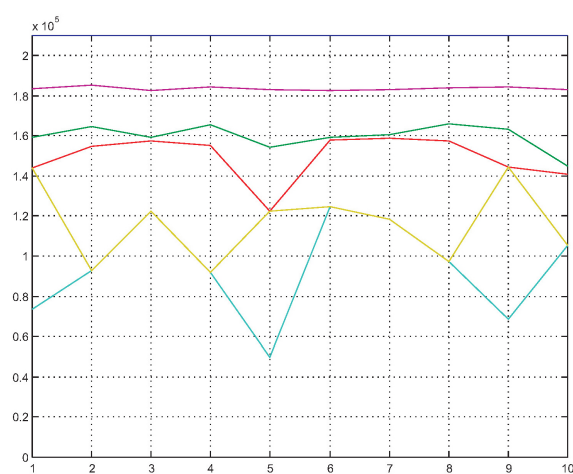
(c) mit Sättigungsgrenze bei 50%



(d) mit Sättigungsgrenze bei 20%



(e) Abweichung des Korrekturvektors



(f) Anzahl berechneter Pixel

Bild A.5: Gray-World Fallstudie 4 [129]



(a) Gray-World Original



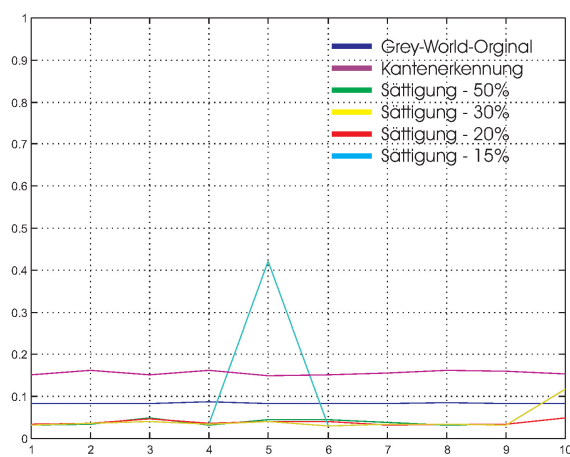
(b) mit Kantenerkennung



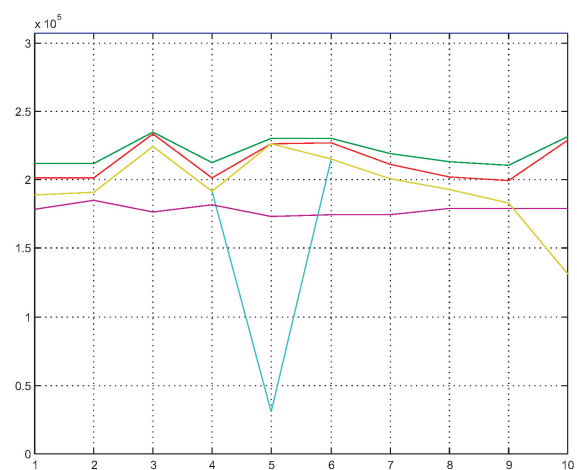
(c) mit Sättigungsgrenze bei 50%



(d) mit Sättigungsgrenze bei 20%



(e) Abweichung des Korrekturvektors



(f) Anzahl berechneter Pixel

Bild A.6: Gray-World Fallstudie 5 [129]

# Anhang B

## Binärisierungsmethoden

### B.0.1 Otsu Methode

Das Verfahren von Otsu wurde im Jahr 1979 von Otsu Nobuyuki publiziert [75]. Dabei handelt es sich um ein Schwellwertverfahren, dessen globaler Schwellwert auf Basis der Varianz eines Bildes ermittelt wird. Wie bereits in Kapitel 3.3 erläutert, setzen derartige Verfahren eine bimodale Grauwertverteilung in Bildern voraus. Die Methode definiert zwei Klassen von Grauwerten  $K_0(T)$  und  $K_1(T)$  sowie die Auftrittswahrscheinlichkeit  $p(x)$  eines Grauwertes  $g$ , mit  $g \in [0; G]$ . Für die Zuordnung der Pixel zu den beiden Klassen ergibt sich folgende Wahrscheinlichkeitsverteilung:

$$P_0(T) = \sum_{g=0}^T p(g) \quad (\text{B.1})$$

$$P_1(T) = 1 - P_0(T) \quad (\text{B.2})$$

$$= \sum_{g=T+1}^G p(g) \quad (\text{B.3})$$

Mit den arithmetischen Mittelwerten  $\mu_0, \mu_1$  der Grauwerte der beiden Klassen ergeben sich die Varianzen zu:

$$\sigma_0^2(T) = \sum_{g=0}^T (g - \mu_0)^2 p(g) \quad (\text{B.4})$$

$$\sigma_1^2(T) = \sum_{g=T+1}^G (g - \mu_1)^2 p(g) \quad (\text{B.5})$$

Das Otsu Verfahren strebt eine Maximierung der Varianz zwischen den Klassen und gleichzeitig eine Minimierung der klasseninternen Varianzen an. Die Varianz zwischen den Klassen berechnet sich durch:

$$\sigma_{zw}^2(T) = P_0(T) \cdot (\mu_0 - \mu)^2 + P_1(T) \cdot (\mu_1 - \mu)^2 \quad (\text{B.6})$$



Dementsprechend ergibt sich die Varianz innerhalb der Klassen aus der Summe der beiden Einzelvarianzen:

$$\sigma_{in}^2(T) = P_0(T) \cdot \sigma_0^2(T) + P_1(T) \cdot \sigma_1^2(T) \quad (\text{B.7})$$

Nach Otsu wird der Schwellwert  $T$  so gewählt, dass der Quotient der Varianzen  $\frac{\sigma_{zw}^2(T)}{\sigma_{in}^2(T)}$  maximal ist.

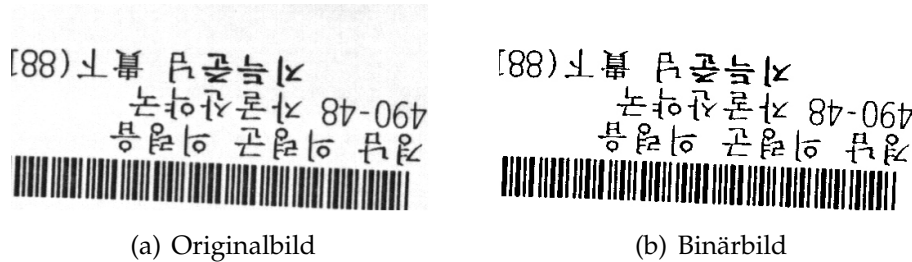


Bild B.1: Binärisierung nach dem Verfahren von Otsu

Bild B.1 zeigt ein Beispiel mit einer bimodalen Grauwertverteilung, für welches das Verfahren einen geeigneten Schwellwert findet. Anders verhält es sich für das Beispiel

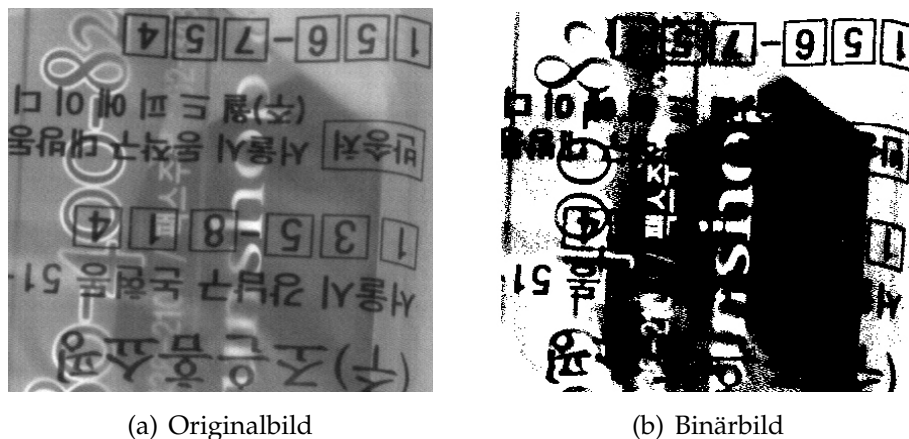


Bild B.2: Binärisierung nach dem Verfahren von Otsu

in Bild B.2. Der unruhige Hintergrund verhindert eine adäquate Trennung der Objekte durch einen globalen Schwellwert.

## B.0.2 Niblack's Verfahren

Wayne Niblack stellte 1986 ein lokal adaptives Binärisierungsverfahren vor. Die Berechnung des Schwellwerts  $T$  für ein Pixel errechnet sich aus dem Mittelwert  $\mu$  und der Stan-

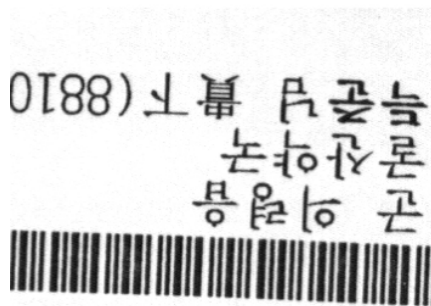
Standardabweichung  $\sigma$  der Grauwerte  $g$  einer  $N \times N$  großen Pixelumgebung:

$$T(x, y) = \mu(x, y) + k * \sigma(x, y) \quad (\text{B.8})$$

$$\mu(x, y) = \frac{1}{N^2} \cdot \sum_{i=-\frac{N-1}{2}}^{\frac{N-1}{2}} \cdot \sum_{j=-\frac{N-1}{2}}^{\frac{N-1}{2}} g(x+i, y+j)$$

$$\sigma(x, y) = \sqrt{\frac{1}{N^2} \sum_{i=-\frac{N-1}{2}}^{\frac{N-1}{2}} \cdot \sum_{j=-\frac{N-1}{2}}^{\frac{N-1}{2}} [g(x+i, y+j) - \mu(x, y)]^2}$$

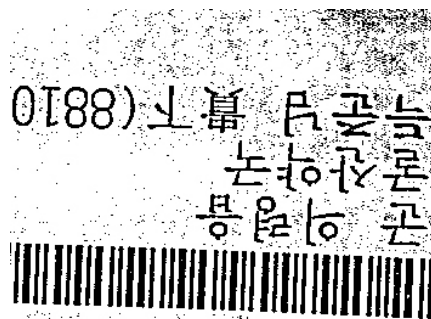
Die Koppelweite  $N$  muss ungerade sein, demzufolge das betrachtete Pixel stets im Mittelpunkt der Maske ist. Folglich muss zur Binärisierung eines Bildes die Maske einmal über das ganze Bild verschoben werden. Der Parameter  $k$  in Gleichung B.9 wird als Niblack's Konstante bezeichnet. Sie ist maßgeblich an der Schwellwertbildung beteiligt und muss experimentell bestimmt werden. Sind in einem Bild die zu extrahierenden Codes dunkler als der Hintergrund, so muss  $k$  negativ gewählt werden. Damit wird der Schwellwert ausgehend vom Mittelwert der Intensitätsverteilung in Abhängigkeit von der Standardabweichung in Richtung niedrigerer Intensitätswerte verschoben.



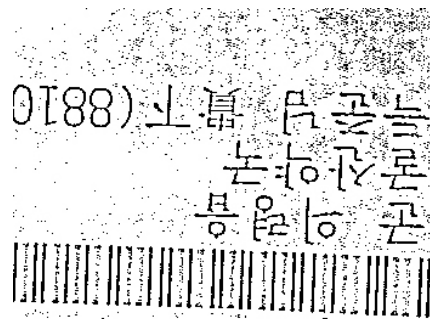
(a) Originalbild



(b)  $k = 0.24$



(c)  $k = -0.24$



(d)  $k = -0.7$

Bild B.3: Binärisierung nach dem Verfahren von Niblack

Die Bilder B.3 und B.4 zeigen an zwei Beispielen den Einfluss von  $k$  bei einer Maskengröße von  $7 \times 7$  Bildpunkten. Vor allem in dem zweiten Beispiel zeigt sich eine deutliche

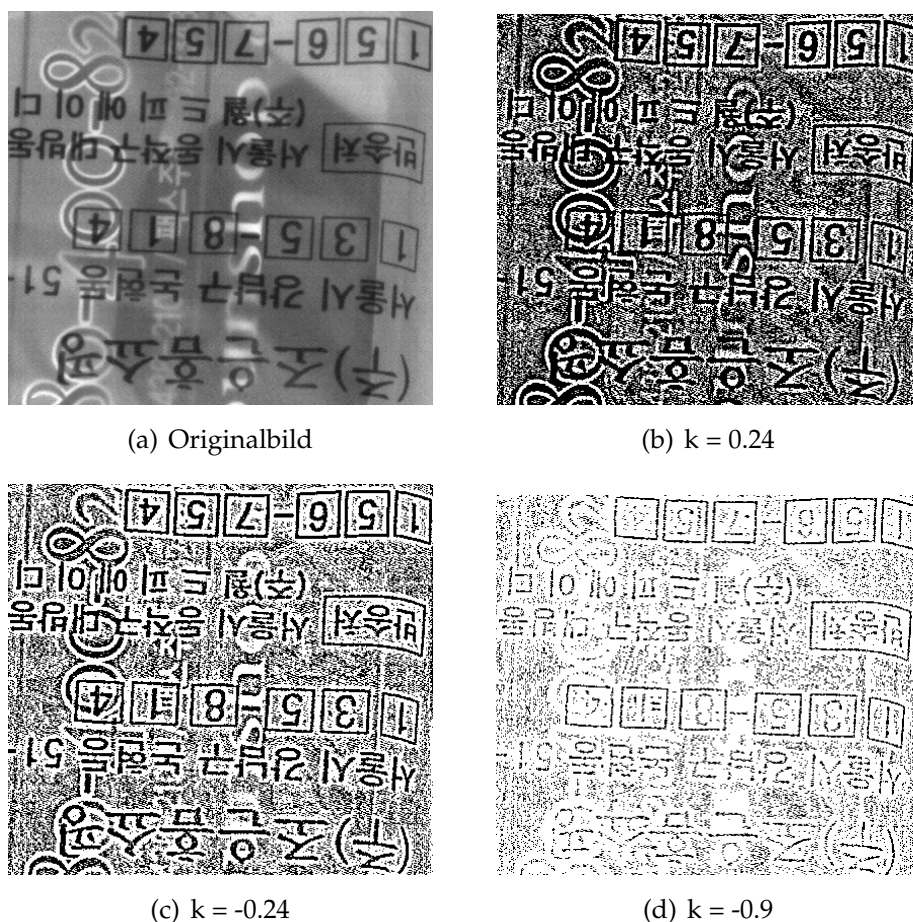


Bild B.4: Binärisierung nach dem Verfahren von Niblack

Verbesserung gegenüber den globalen Schwellwertverfahren. Auffällig ist jedoch, dass in Bildregionen ohne Text das Rauschen durch die Binärisierung deutlich zunimmt. Dieser Effekt resultiert aus sehr niedrigen Streuungen in solchen Bereichen, so dass sich die dazugehörigen Schwellwerte sehr nah an den Mittelwerten befinden.

### B.0.3 Sauvola Methode

Sauvolas Methode basiert auf Niblack's Verfahren und bezweckt primär eine Verbesserung des Rauschverhaltens. Die Berechnung der Schwellwerte erfolgt in Abhängigkeit eines zusätzlichen Parameters  $D$ , der die dynamische Reichweite der Standardabweichungen repräsentiert:

$$T(x, y) = \mu(x, y) * \left[ 1 - k * \left( 1 - \frac{\sigma(x, y)}{D} \right) \right]. \quad (\text{B.9})$$

Dabei ist die Verschiebung des Schwellwerts, ausgehend vom Mittelwert der Verteilung, antiproportional zur Standardabweichung. Abbildung B.5a zeigt, dass die Schwellwerte der beiden Verfahren für Regionen mit Text sehr ähnlich sind. Die Verbesserung durch Anwendung des Sauvola Algorithmus wird in Bild B.5b deutlich, dass ausschließlich aus



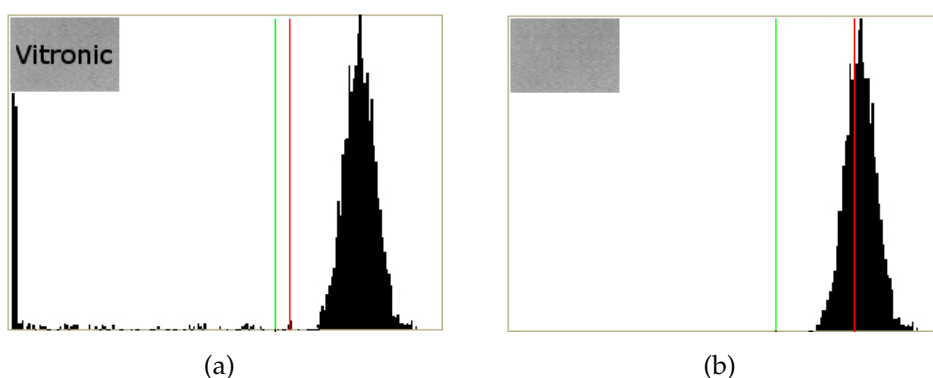


Bild B.5: Vergleich der Schwellwertberechnung nach Niblack (rot) und Sauvola (grün)

Hintergrund besteht. Hier liegt der Schwellwert deutlich unter dem von Niblack, so dass der Hintergrund vollständig als solcher erkannt wird. Die Leistungsfähigkeit des Ver-

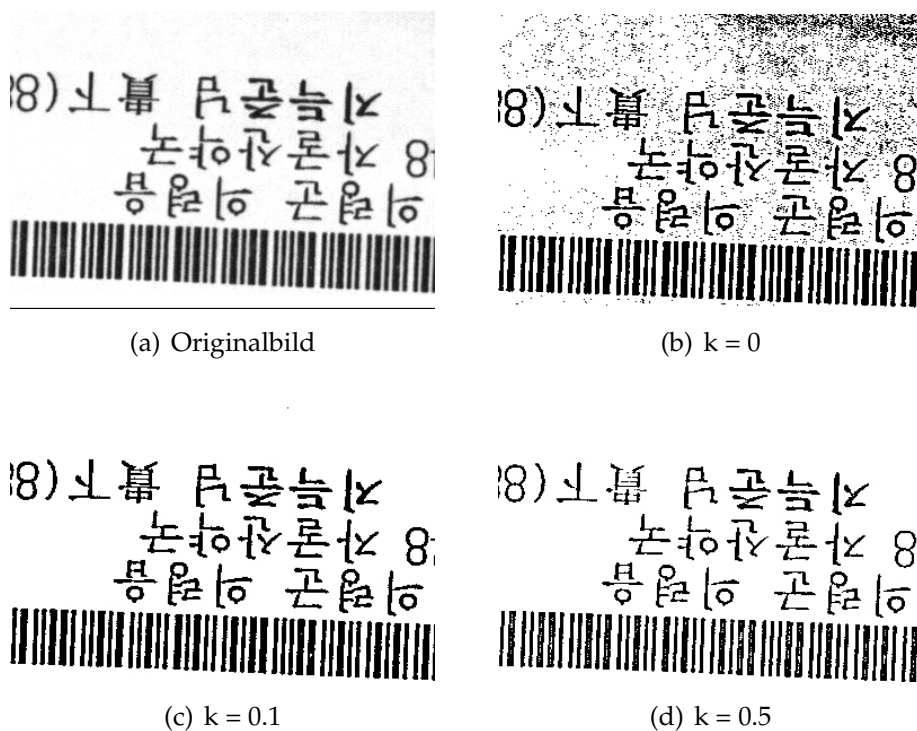
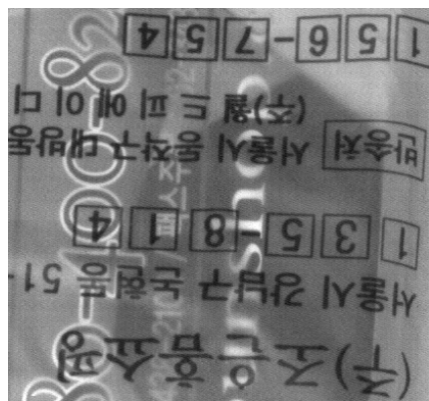
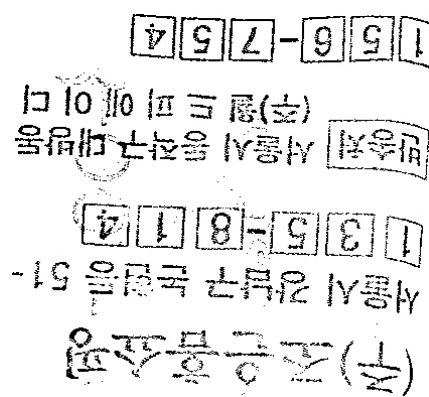


Bild B.6: Binärisierung nach dem Verfahren von Sauvola mit  $N = 7$  und  $D = 128$

fahrens wird in den Bildern B.6, B.7 veranschaulicht. Anhand dieser Bilder wird deutlich, dass das Verfahren sowohl für Bilder mit geringem Rauschen als auch für Bilder mit variierenden Hintergründen sehr gute Ergebnisse liefert. Mit größeren  $k$  werden Hintergrund und Text stärker voneinander getrennt, jedoch führt dies auch zu einer Verdünnung der Textkonturen.



(a) Originalbild

(b)  $k = 0$ (c)  $k = 0.05$ (d)  $k = 0.24$ Bild B.7: Binärisierung nach dem Verfahren von Sauvola mit  $N = 7$  und  $D = 128$

# Anhang C

## Ergänzungen der Entwurfsautomatisierung

### C.0.4 BNF-Notationen

In Kapitel 4.3 wurde die Syntax der IP-Core Metadaten vorgestellt. Zur Beschreibung dieser kontextfreien Grammatik wurde die Backus-Naur-Form (BNF) herangezogen. Abbil-

```
<file> ::= { <core> }
<core> ::= identifier
        "{
            "Name"    "=" identifier ";"
            "ID"       "=" integer ";"
            { [ <data> ";" ] }
        }"
<data> ::= "Data-In" integer          "=" <type> <domain> |
        "Data-Out"                    "=" <type> <domain> |
        "Clock"                       "=" integer ":" integer ["MHz"] |
        "Type"                         "=" <output_type> |
        "Dependencies"                 "=" <set> |
        "Equivalents"                  "="
        "{
            integer | <sequence>
            { ", " ( integer | <sequence> ) }
        }" |
        "Latency"                      "=" integer |
        "Datasize-Required"             "=" integer |
        "Bitstream-Size"                "=" integer |
        "Compatible-Slot-IDs"           "=" <set> |
        "Datasize-Factor"              "=" integer [ "/" integer]
```

Bild C.1: BNF Metadatenmodell [123]



Abbildung C.1 zeigt die BNF-Notationen der Metadaten. Leerzeichen und Zeilentrenner werden bei der Auswertung ignoriert, sichtbare Zeichen zählen zu den so genannten Terminalsymbolen. Vertikale Striche dienen der Angabe von Alternativen bei der Definition syntaktischer Variablen (blau dargestellt). Während einer Syntaxanalyse wird der Text der Metadaten auf die syntaktische Variable höchster Ebene (hier: <file>) zurückgeführt. Abbildung C.2 zeigt zusätzlich benötigte Definitionen. Die BNF-Notation, die der Spezi-

```

<type>          ::= "Integer" | "Bitvector"
<domain>        ::= <set> | <interval>
<set>           ::= "{" integer { ",", integer } "}"
<interval>      ::= "[" integer [ ",", integer ] "]"
<sequence>      ::= "(" integer { ",", integer } ")"
<output_type>   ::= "Input" | "Output" | "Input_Output"

```

Bild C.2: BNF Metadatenmodell – Typendefinitionen [123]

```

<file>          ::= { <set> }
<set>           ::= identifier "{" { <core> ";" } "}"
<core>          ::= identifier "(" <listOfPar> ")"
<listOfPar>     ::= [ integer | <core> { ",", integer | <core> } ]

```

Bild C.3: BNF Modell der Spezifikationssemantik [123]

fikationssemantik zugrunde liegt, ist in Abbildung C.3 dargestellt. Die rekursive Struktur ermöglicht das vorgestellte Prinzip der Verschachtelung von IP-Cores in der Spezifikation. Auf diese Weise kann ein <core> eine Zahl oder einen weiteren <core> in der Parameterliste enthalten.

## C.0.5 Scheduling Algorithmus

Bestimmung der Schedulingparameter für dynamisch rekonfigurierbare Schaltungen:

- Maschinenumgebung: eine beliebige Anzahl von parallelen Maschinen ('R')
- Verarbeitungseigenschaften: Jobs sind unteilbar und haben Präzedenzbeziehungen zueinander
- Optimierungskriterium: Minimierung der gesamten Verarbeitungszeit (Kostenfunktion :  $C_{max}$ ).

Daraus ergibt sich die formelle Notation:  $(R \mid prec \mid C_{max})$ .

Unterschiedliche Schedulingmethoden sind für derartige Aufgaben einsetzbar. Das im Folgenden vorgestellte Verfahren ist von dem *List-Scheduling* abgeleitet [16]. Hierzu werden diverse Notationen eingeführt:

- $J_1, \dots, J_N : N$  Verarbeitungen, bestehend aus zwei Teilen:  $J_{i,a} \Rightarrow$  Rekonfiguration, Parametrierung des Cores;  $J_{i,b} \Rightarrow$  Ausführung
- $M_1, \dots, M_M : M$  verfügbare Maschinen
- $\mu_1, \dots, \mu_N : \text{Maschinenzuweisung für jeden Job } (\mu \in M)$
- $\phi(J) : \text{Startzeit des Jobs } J$
- $\delta(J) : \text{Dauer des Jobs } J \text{ (in Taktzyklen)}$
- $V : \text{Liste aller ungeplanten Jobs}$
- $V_{ready} : \text{Liste der Jobs, deren vorrangige Jobs bereits bearbeitet wurden}$

Folgende zusätzliche Bedingung sind notwendig:

- Es existieren zwei Arten Präzedenzbeziehungen: die zwischen zwei Teilen einer Verarbeitung ( $J_i = J_{i,a} \rightarrow J_{i,b}$ ) und die zwischen zwei Ausführungsjobs

$$\{\text{Präzedenzbeziehungen}\} \subset \{J_{i,a} \rightarrow J_{i,b} \mid \forall i\} \cup \{J_{i,b} \rightarrow J_{j,b} \mid \forall (i, j) \in [1, N]^2\}$$

- Die Rekonfiguration und die Ausführung eines Jobs finden auf derselben Maschine statt
- Rekonfigurationen können nicht gleichzeitig ausgeführt werden

$$\forall (i, j) \in [1, N]^2, i \neq j, [\phi(J_{i,a}), \phi(J_{i,a}) + \delta(J_{i,a})] \cap [\phi(J_{j,a}), \phi(J_{j,a}) + \delta(J_{j,a})] = \emptyset$$

- Maschinen können sich unterscheiden. Jede Maschine kann mehrere Jobs ausführen, jedoch kann ein Job nur auf einer Maschine ausgeführt werden

$$\forall i \in [1, N], \mu_i \subseteq \{M_1, \dots, M_M\}$$

$$\forall (i, j) \in [1, N]^2, i \neq j, \mu_i \cap \mu_j = \emptyset \vee \mu_i = \mu_j$$

Der *List-Scheduling* Algorithmus beruht auf der Berechnung einer Prioritätenliste. Als Dringlichkeitskriterium wird die maximale Zeit (in Taktzyklen) bis zum Ende der Verarbeitungskette gewählt. Ergeben sich mehrere gleichwertige Lösungen, so sind die Ausführungsjobs ( $J_{i,b}$ ) vor den Rekonfigurationsjobs vorzuziehen. Für den Scheduling-Algorithmus werden folgende Funktionen benutzt:

- $freeze(J) : \text{reserviert eine Maschine, die } J \text{ ausführen kann}$

- $unfreeze(J)$  : gibt die reservierte Maschine für Job  $J$  frei
- $unused(J)$  : liefert die Anzahl verfügbare Maschinen, die den Job  $J$  ausführen können (wenn  $J_{i,a}$  aktiv ist, gilt für jedes  $j$ :  $unused(J_{j,a}) = 0$ ).
- $sort(V)$  : ordnet die Menge  $V$  hinsichtlich ihrer Priorität
- $fanin(J)$  : liefert die Menge der Vorgänger von  $J_i$

```

c := 1 ;           /* aktuelles Zeitintervall */
Vsched := ∅ ;    /* Menge der Operationen in Verarbeitung */

while (V ≠ ∅) do

    /* Ressourcen freigeben */
    for all J ∈ Vsched
        if (c = φ(J) + δ(J))
            unfreeze(J);
            Vsched := Vsched \ {J};
        end if;
    end for;

    /* Ressourcen allokieren */
    Vready = {J | fanin(J) ∩ V = ∅ ∧ J ∉ Vsched };
    Vsort = sort(Vready);

    for all J ∈ Vsort
        if unused(J) > 0
            φ(J) := c ;
            freeze(J);
            Vsched := Vsched ∪ {J};
            V := V \ {J};
        end if;
    end for;

    /* Zeitintervall inkrementieren */
    c := min{φ(J) + δ(J) | J ∈ Vsched};

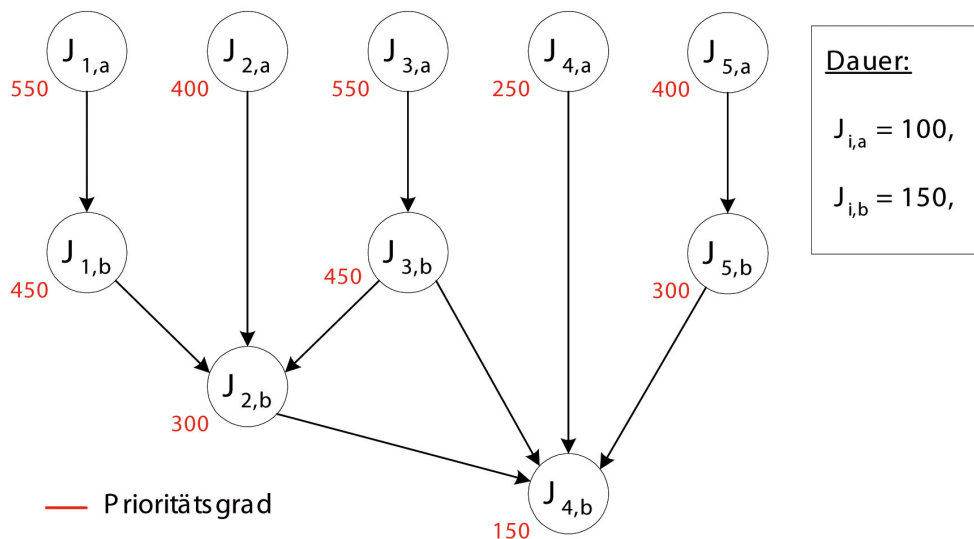
end while ;

```

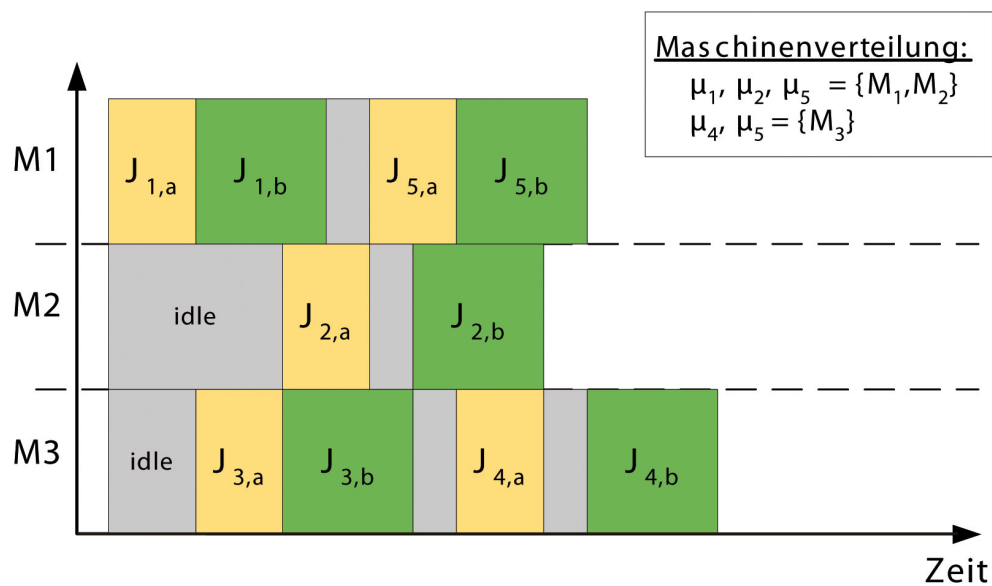
Bild C.4: Scheduling Algorithmus [123]



Ein entsprechender Beispielgraph eines Ablaufplans und die resultierende Aufteilung der IP-Cores auf die zur Verfügung stehenden Slots (Maschinen) ist in Abbildung C.5 dargestellt.



(a) Beispielgraph



(b) Resultierende Job-Verteilung

Bild C.5: Scheduling Beispiel



# Literaturquellen

- [1] ALTERA,CORP. Using the logiclock methodology in the quartus ii design software. *Application Note 161*, 2003.
- [2] ALTERA,CORP. Lowering the total cost of ownership in industrial applications. *White Paper*, <http://www.altera.com>, 2010.
- [3] S. BABVEY, J. A. FERNÁNDEZ-ZEPEDA, A. G. BOURGEOIS, and S. W. McLAUGHLIN. An efficient r-mesh implementation of ldpc codes message-passing decoder. In *19th International Parallel and Distributed Processing Symposium (IPDPS)*, 2005.
- [4] J. W. BACKUS, F. L. BAUER, J. GREEN, C. KATZ, J. MCCARTHY, A. J. PERLIS, H. RUTISHAUSER, K. SAMELSON, B. VAUQUOIS, J. H. WEGSTEIN, A. VAN WIJNGAARDEN, M. WOODGER, and P. NAUR. Revised report on the algorithm language algol 60. *Commun. ACM*, 6(1):1–17, 1963.
- [5] K. BARNARD, V. C. CARDEI, and B. V. FUNT. A comparison of computational color constancy algorithms. i: Methodology and experiments with synthesized data. *IEEE Transactions on Image Processing*, 11(9):972–984, 2002.
- [6] V. BAUMGARTE, G. EHLERS, F. MAY, A. NÜCKEL, M. VORBACH, and M. WEINHARDT. Pact xpp—a self-reconfigurable data processing architecture. *J. Supercomput.*, 26(2):167–184, 2003.
- [7] J. BECKER, A. DONLIN, and M. HÜBNER. New tool support and architectures in adaptive reconfigurable computing. In *IFIP WG 10.5 International Conference on Very Large Scale Integration of System-on-Chip, VLSI-SoC*, pages 134–139, 2007.
- [8] P. BELLOWS and B. HUTCHINGS. Jhdl - an hdl for reconfigurable systems. In *FCCM '98: Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines*, page 175, Washington, DC, USA, 1998. IEEE Computer Society.
- [9] K. BENKRID and D. CROOKES. From application descriptions to hardware in seconds: a logic-based approach to bridging the gap. *IEEE Trans. VLSI Syst.*, 12(4):420–436, 2004.
- [10] P. BENOIT, G. SASSATELLI, L. TORRES, D. DEMIGNY, M. ROBERT, and G. CAMBON. Metrics for reconfigurable architectures characterization: Remanence and scalability. In *IPDPS '03: Proceedings of the 17th International Symposium on Parallel and Distributed Processing*, page 176.1, Washington, DC, USA, 2003. IEEE Computer Society.
- [11] C. BOBDA. *Introduction to Reconfigurable Computing: Architectures, Algorithms, and Applications*. Springer Publishing Company, Incorporated, 2007.
- [12] C. BOBDA, A. AHMADINIA, M. MAJER, J. TEICH, S. P. FEKETE, and J. VAN DER VEEN. Dynoc: A dynamic infrastructure for communication in dynamically reconfigurable devices. *CoRR*, abs/cs/0510039, 2005.

- [13] C. BOBDA, M. MAJER, A. AHMADINIA, T. HALLER, A. LINARTH, and J. TEICH. The erlangen slot machine: Increasing flexibility in fpga-based reconfigurable platforms. *Proceedings of the 2005 IEEE International Conference on Field-Programmable Technology, FPT*, pages 37–42, 2005.
- [14] C. BOBDA, M. MAJER, D. KOCH, A. AHMADINIA, and J. TEICH. A dynamic noc approach for communication in reconfigurable devices. In *Field Programmable Logic and Application, 14th International Conference, FPL*, pages 1032–1036, 2004.
- [15] G. BORGEFORS. Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, 34(3):344 – 371, 1986.
- [16] P. BRUCKER. *Scheduling Algorithms*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.
- [17] J. M. CARVER, R. N. PITTMAN, and A. FORIN. Automatic bus macro placement for partially reconfigurable fpga designs. In *FPGA '09: Proceeding of the ACM/SIGDA international symposium on Field programmable gate arrays*, pages 269–272, New York, NY, USA, 2009. ACM.
- [18] CELOXICA. Real world experiences designing for mixed cpu + fpga systems. *White Paper*, <http://www.celoxica.com>, 2002.
- [19] C. CHANG, J. WAWRZYNEK, and R. W. BRODERSEN. Bee2: A high-end reconfigurable computing system. *IEEE Des. Test*, 22(2):114–125, 2005.
- [20] M. COLE. *Algorithmic skeletons: structured management of parallel computation*. MIT Press, Cambridge, MA, USA, 1991.
- [21] S. D. CRAVEN. Structured approach to dynamic computing application development. In *Dissertation thesis, Bradley Department of Electrical and Computer Engineering, Blacksburg, Virginia*, 2008.
- [22] D. DEHON, J. ADAMS, M. DELORIMIER, N. KAPRE, Y. MATSUDA, H. NAEIMI, M. VANIER, and M. WRIGHTON. Design patterns for reconfigurable computing. In *FCCM '04: Proceedings of the 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, pages 13–23, Washington, DC, USA, 2004. IEEE Computer Society.
- [23] L. DI STEFANO, S. MATTOCCIA, and M. MOLA. An efficient algorithm for exhaustive template matching based on normalized cross correlation. In *ICIAP '03: Proceedings of the 12th International Conference on Image Analysis and Processing*, page 322, Washington, DC, USA, 2003. IEEE Computer Society.
- [24] F. DITTMANN. Algorithmic skeletons for the programming of reconfigurable systems. In *Software Technologies for Embedded and Ubiquitous Systems, 5th IFIP WG 10.2 International Workshop, SEUS*, pages 358–367, 2007.
- [25] F. DITTMANN, S. FRANK, and S. OBERTHÜR. Algorithmic skeletons for the design of partially reconfigurable systems. In *22nd IEEE International Symposium on Parallel and Distributed Processing, IPDPS*, pages 1–8, 2008.
- [26] F. DITTMANN, A. RETTBERG, and F. SCHULTE. A y-chart based tool for reconfigurable system design. In *18th International Conference on Architecture of Computing Systems, ARCS Workshops*, pages 67–73, 2005.
- [27] F. DITTMANN, A. RETTBERG, and R. WEBER. Path concepts for a reconfigurable bit-serial synchronous architecture. In *Embedded and Ubiquitous Computing - EUC 2005, International Conference EUC*, pages 448–457, 2005.



- [28] B. DONCHEV, G. KUZMANOV, and G. N. GAYDADJIEV. External memory controller for virtex ii pro. In *Proceedings of International Symposium on System-on-Chip*, pages 37–40, 2006.
- [29] H. EECKHAUT, H. DEVOS, P. LAMBERT, D. D. SCHRIJVER, W. V. LANCKER, V. NOLLET, P. AVASARE, T. CLERCKX, F. VERDICCHIO, M. CHRISTIAENS, P. SCHELKENS, R. V. DE WALLE, and D. STROOBANDT. Scalable, wavelet-based video: From server to hardware-accelerated client. *IEEE Transactions on Multimedia*, 9(7):1508–1519, 2007.
- [30] G. ESTRIN. Organization of computer systems: the fixed plus variable structure computer. In *IRE-AIEE-ACM '60 (Western): Papers presented at the May 3-5, 1960, western joint IRE-AIEE-ACM computer conference*, pages 33–40, New York, NY, USA, 1960. ACM.
- [31] A. FLYNN, A. GORDON-ROSS, and A. D. GEORGE. Bitstream relocation with local clock domains for partially reconfigurable fpgas. In *Design, Automation and Test in Europe, DATE*, pages 300–303, 2009.
- [32] D. FORSYTH. Sampling, resampling and colour constancy. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:1300, 1999.
- [33] W. T. FREEMAN and D. H. BRAINARD. Bayesian decision theory, the maximum local mass estimate, and color constancy. In *ICCV*, pages 210–217, 1995.
- [34] D. GAJSKI and R. H. KUHN. New vlsi tools - guest editors' introduction. *IEEE Computer*, 16(12):11–14, 1983.
- [35] D. GÖHRINGER, M. MAJER, and J. TEICH. Bridging the gap between relocatability and available technology: The erlangen slot machine. In P. M. ATHANAS, J. BECKER, G. BREBNER, and J. TEICH, eds., *Dynamically Reconfigurable Architectures*, number 06141 in Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2006. <<http://drops.dagstuhl.de/opus/volltexte/2006/736>> [date of citation: 2006-01-01].
- [36] R. C. GONZALEZ and R. WOODS. *Digital Image Processing, 3rd ed.* Prentice Hall, 2008.
- [37] F. GRAHAM. Coefficient colour constancy. In *Dissertationsarbeit, Simon Fraser University*, 1995.
- [38] S. GUCCIONE and D. LEVI. Run-time parameterizable cores. In *FPL '99: Proceedings of the 9th International Workshop on Field-Programmable Logic and Applications*, pages 215–222, London, UK, 1999. Springer-Verlag.
- [39] B. K. GUNTURK, J. GLOTZBACH, Y. ALTUNBASAK, R. W. SCHAFFER, and R. M. MERSEREAU. Demosaicking: color filter array interpolation. *Signal Processing Magazine*, 22:44–54, 2005.
- [40] R. W. HARTENSTEIN, M. HERZ, T. HOFFMANN, and U. NAGELDINGER. On reconfigurable co-processing units. In *IPPS/SPDP Workshops*, pages 67–72, 1998.
- [41] R. W. HARTENSTEIN, R. KRESS, and H. REINIG. A reconfigurable data-driven alu for xputers. In *IEEE Workshop on FPGAs for Custom Computing Machines, FCCM*. IEEE, 1994.
- [42] M. HARTMANN, V. PANTAZIS, T. V. AA, M. BEREKOVIC, C. HOCHBERGER, and B. D. SUTTER. Still image processing on coarse-grained reconfigurable array architectures. In *Proceedings of the 2007 5th Workshop on Embedded Systems for Real-Time Multimedia, ESTMedia*, pages 67–72. IEEE, 2007.
- [43] M. HÜBNER. Dynamisch und partiell rekonfigurierbare hardware-systemarchitektur mit echtzeitfähiger on-demand funktionalität. In *Dissertationsarbeit, Universität Karlsruhe (TH)*, 2007.
- [44] M. HÜBNER, T. BECKER, and J. BECKER. Real-time lut-based network topologies for dynamic and partial fpga self-reconfiguration. In *Proceedings of the 17th Annual Symposium on Integrated Circuits and Systems Design, SBCCI*, pages 28–32, 2004.

- [45] J. HE, Q. D. M. DO, A. C. DOWNTON, and J. H. KIM. A comparison of binarization methods for historical archive documents. In *ICDAR '05: Proceedings of the Eighth International Conference on Document Analysis and Recognition*, pages 538–542, Washington, DC, USA, 2005. IEEE Computer Society.
- [46] S. HEITHECKER and R. ERNST. Traffic shaping for an fpga based sdram controller with complex qos requirements. In *DAC '05: Proceedings of the 42nd annual Design Automation Conference*, pages 575–578, New York, NY, USA, 2005. ACM.
- [47] S. HEITHECKER, A. C. LUCAS, and R. ERNST. A mixed qos sdram controller for fpga-based high-end image processing. In *Workshop on Signal Processing Systems Design and Implementation (SIPS), IEEE*, page 11, 2003.
- [48] R. H. HIBBARD. Apparatus and method for adaptively interpolating a full color image utilizing luminance gradients. *U.S. Patent 5,382,976*, 1995.
- [49] K. HIRAKAWA and T. W. PARKS. Adaptive homogeneity-directed demosaicing algorithm. In *ICIP (3)*, pages 669–672, 2003.
- [50] S. D. HORDLEY and G. D. FINLAYSON. Re-evaluating colour constancy algorithms. In *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 1*, pages 76–79, Washington, DC, USA, 2004. IEEE Computer Society.
- [51] M. HÜBNER, C. SCHUCK, M. KÜHNLE, and J. BECKER. New 2-dimensional partial dynamic reconfiguration techniques for real-time adaptive microelectronic circuits. In *IEEE Computer Society Annual Symposium on VLSI, ISVLSI*, pages 97–102, 2006.
- [52] D. P. HUTTENLOCHER, G. A. KLANDERMAN, and W. A. RUCKLIDGE. Comparing images using the hausdorff distance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(9):850–863, 1993.
- [53] Y. IWASAKI and Y. KUROI. Real-time robust vehicle detection through the same algorithm both day and night. *Proc. Of the 2007 Int. Conference on Wavelet Analysis and Pattern Recognition (ICWAPR)*, 3:1008–1014, 2007.
- [54] B. JACKSON. Partial reconfiguration design with planahead 9.2. *Technical report, Xilinx, Inc.*, 2008.
- [55] B. JÄHNE. *Digitale Bildverarbeitung*. Springer, 6., überarb. u. erw. aufl. edition, 2005.
- [56] M. KATEVENIS, S. SIDIROPOULOS, and C. COURCOUBETIS. Weighted round-robin cell multiplexing in a general-purpose atm switch chip. *IEEE Journal on Selected Areas in Communications*, 9(8):1265–1279, 1991.
- [57] E. KELLER. Jroute: A run-time routing api for fpga hardware. In *IPDPS '00: Proceedings of the 15 IPDPS 2000 Workshops on Parallel and Distributed Processing*, pages 874–881, London, UK, 2000. Springer-Verlag.
- [58] J. KENDER. Saturation, hue and normalized color. *Carnegie-Mellon University, Computer Science Dept., Pittsburgh, PA*, 1976.
- [59] S. KOH and O. DIESSEL. Comma: a communications methodology for dynamic module-based reconfiguration of fpgas. In *International Conference on Architecture of Computing Systems, Dynamically Reconfigurable Systems Workshop Proceedings*, pages 173–182, 2006.
- [60] S. KOH and O. DIESSEL. Module graph merging and placement to reduce reconfiguration overheads in paged fpga devices. In *International Conference on Field Programmable Logic and Applications, FPL*, pages 293–298, 2007.
- [61] S. KOH and O. DIESSEL. The effectiveness of configuration merging in point-to-point networks for module-based fpga reconfiguration. In *16th IEEE International Symposium on Field-Programmable Custom Computing Machines, FCCM*, pages 65–76, 2008.

- [62] H. T. KUNG and P. L. LEHMAN. Systolic (vlsi) arrays for relational database operations. In *SIGMOD '80: Proceedings of the 1980 ACM SIGMOD international conference on Management of data*, pages 105–116, New York, NY, USA, 1980. ACM.
- [63] E. H. LAND. Lightness and the retinex theory. *J. Opt. Soc. Am.*, 61:1–11, 1971.
- [64] C. A. LAROCHE and M. A. PRESCOTT. Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients. *U.S. Patent 5,373,322*, 1994.
- [65] M.-H. LEE, H. SINGH, G. LU, N. BAGHERZADEH, F. J. KURDAHI, E. M. C. FILHO, and V. C. ALVES. Design and implementation of the morphosys reconfigurable computing processor. *J. VLSI Signal Process. Syst.*, 24(2/3):147–164, 2000.
- [66] B. LEIBE, A. LEONARDIS, and B. SCHIELE. Combined object categorization and segmentation with an implicit shape model. In *Workshop on Statistical Learning in Computer Vision, ECCV*, 2004.
- [67] J. P. LEWIS. Fast normalized cross-correlation. *Canadian Image Processing and Pattern Recognition Society*, 1995.
- [68] D. G. LOWE. Local feature view clustering for 3d object recognition. In *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 682–688, 2001.
- [69] T. MAKIMOTO. The rising wave of field programmability. In *Field-Programmable Logic and Applications, The Roadmap to Reconfigurable Computing, 10th International Workshop, FPL*, pages 1–6, 2000.
- [70] S. MCMILLAN and S. GUCCIONE. Partial run-time reconfiguration using jrtr. In *FPL '00: Proceedings of the The Roadmap to Reconfigurable Computing, 10th International Workshop on Field-Programmable Logic and Applications*, pages 352–360, London, UK, 2000. Springer-Verlag.
- [71] B. MEI, S. VERNALDE, D. VERKEST, and R. LAUWEREINS. Design methodology for a tightly coupled vliw/reconfigurable matrix architecture: A case study. In *DATE '04: Proceedings of the conference on Design, automation and test in Europe*, page 21224, Washington, DC, USA, 2004. IEEE Computer Society.
- [72] B. MEI, S. VERNALDE, D. VERKEST, H. D. MAN, and R. LAUWEREINS. Adres: An architecture with tightly coupled vliw processor and coarse-grained reconfigurable matrix. In *In Field-Programmable Logic and Applications, FPL*. IEEE, 2003.
- [73] G. E. MOORE. Vlsi: Some fundamental challenges. *IEEE spectrum*, pages 30–37, 1979.
- [74] W. NIBLACK. *An Introduction to Digital Image Processing*. Prentice Hall, 1986.
- [75] N. OTSU. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, January 1979.
- [76] PACT. Programming xpp-iii processors - release 2.0.1. *White Paper*, <http://www.pactxpp.com>, 2006.
- [77] PACT. Xpp-iii processor overview - release 2.0.1. *White Paper*, <http://www.pactxpp.com>, 2006.
- [78] K. PAPADIMITRIOU, A. ANYFANTIS, and A. DOLLAS. Methodology and experimental setup for the determination of system-level dynamic reconfiguration overhead. In *FCCM '07: Proceedings of the 15th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, pages 335–336, Washington, DC, USA, 2007. IEEE Computer Society.
- [79] C. PATTERSON and S. A. GUCCIONE. Jbits design abstractions. In *FCCM '01: Proceedings of the 9th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, pages 251–252, Washington, DC, USA, 2001. IEEE Computer Society.
- [80] J. B. PETER, P. ATHANAS, and E. KELLER. Java debug hardware models using jbits, 2001.

- [81] S. PILLEMENT, O. SENTIEYS, and R. DAVID. Dart: a functional-level reconfigurable architecture for high energy efficiency. *EURASIP J. Embedded Syst.*, 2008:1–13, 2008.
- [82] T. PIONTECK, C. ALBRECHT, R. KOCH, T. BRIX, and E. MAEHLE. Design and simulation of runtime reconfigurable systems. In *DDECS '08: Proceedings of the 2008 11th IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems*, pages 1–4, Washington, DC, USA, 2008. IEEE Computer Society.
- [83] T. PIONTECK, C. ALBRECHT, R. KOCH, E. MAEHLE, M. HÜBNER, and J. BECKER. Communication architectures for dynamically reconfigurable fpga designs. In *21th International Parallel and Distributed Processing Symposium (IPDPS)*, pages 1–8, 2007.
- [84] PULNIX AMERICA, INC. Specifications of the camera link interface standard for digital cameras and frame grabbers. *Standard Specification*, <http://www.machinevisiononline.org/>, 2000.
- [85] B. RADUNOVIC. An overview of advances in reconfigurable computing systems. In *HICSS*, 1999.
- [86] F. J. RAMMIG. A concept for the editing of hardware resulting in an automatic hardware-editor. In *DAC '77: Proceedings of the 14th Design Automation Conference*, pages 187–193, Piscataway, NJ, USA, 1977. IEEE Press.
- [87] G. SASSATELLI, L. TORRES, P. BENOIT, T. GIL, C. DIOU, G. CAMBON, and J. GALY. Highly scalable dynamically reconfigurable systolic ring-architecture for dsp applications. In *DATE '02: Proceedings of the conference on Design, automation and test in Europe*, page 553, Washington, DC, USA, 2002. IEEE Computer Society.
- [88] J. J. SAUVOLA and M. PIETIKÄINEN. Adaptive document image binarization. *Pattern Recognition*, 33(2):225–236, 2000.
- [89] A. SIKORA. *Programmierbare Logikbauelemente*. Carl Hanser Verlag, 2001.
- [90] S. K. SINGH, D. S. CHAUHAN, M. VATSA, and R. SINGH. A robust skin color based face detection algorithm, tamkang. *Journal of Science and Engineering*, 6:227–234, 2003.
- [91] L. D. STEFANO and E. VIARANI. Vehicle detection and tracking using the block matching algorithm. In *Proc. of 3rd IMACS/IEEE*, pages 4491–4496, 1999.
- [92] R. STEINBRECHER. *Bildverarbeitung in der Praxis*. Wien: Oldenbourg, 2002.
- [93] R. STEINMETZ. *Multimedia-Technologie : Grundlagen, Komponenten und Systeme*. Springer, Berlin, 3. edition, 1999.
- [94] M. SUZUKI, Y. HASEGAWA, V. M. TUAN, S. ABE, and H. AMANO. A cost-effective context memory structure for dynamically reconfigurable processors. In *20th International Parallel and Distributed Processing Symposium (IPDPS)*, 2006.
- [95] M. SUZUKI, Y. HASEGAWA, Y. YAMADA, N. KANEKO, K. DEGUCHI, and H. AMANO. Stream applications on the dynamically reconfigurable processor. *International Conference on Field-Programmable Technology, FPT*, 2004.
- [96] R. VAIDYANATHAN and J. L. TRAHAN. *Dynamic Reconfiguration: Architectures and Algorithms (Series in Computer Science (Kluwer Academic/Plenum Publishers).)*. Plenum Publishing Co., 2004.
- [97] J. VILLASENOR, C. JONES, and B. SCHONER. Video communications using rapidly reconfigurable hardware. *IEEE Transactions on Circuits and Systems for Video Technology*, 5:565–567, 1995.
- [98] VITRONIC GMBH. *viro<sup>WSI</sup> vollautomatische prüfung von schweißnähten*. *Technische Information – Optische Schweissnahtprüfung*, <http://www.vitronic.de>, 2007.

- [99] VITRONIC GMBH. Vipac kamerabasierte identifikationstechnologie. *Technische Information*, <http://www.vitronic.de>, 2009.
- [100] J. E. VOLDER. The cordic trigonometric computing technique. *IRE Transactions on Electronic Computing*, EC-8:330–334, 1959.
- [101] A. WARKENTIN and F. DITTMANN. Data transfer protocols for a two slot based reconfigurable platform. In *Proceedings of the 2nd International Workshop on Reconfigurable Communication-centric Systems-on-Chip, ReCoSoC*, pages 8–15. Univ. Montpellier II, 2006.
- [102] E. WELCH, R. MOORHEAD, and J. OWENS. Image processing using the hsi color space. *Proc.'91 IEEE Southeastcon*, 1991.
- [103] X. WEN, H. YUAN, C. YANG, C. SONG, B. DUAN, and H. ZHAO. Improved haar wavelet feature extraction approaches for vehicle detection. In *Intelligent Transportation Systems Conference (ITSC)*. IEEE, pages 1050–1053, 2007.
- [104] T. WILLIAMS. Hsi conversion brings true color image processing to life. *Comput. Des.*, 27(2):28–29, 1988.
- [105] XILINX, INC. Two flows for partial reconfiguration: Module based or difference based. *Application Note XAPP290*, 2003.
- [106] XILINX, INC. Opb hwicap. *Product Specification, DS280(v1.3)*, 2004.
- [107] XILINX, INC. Virtex-4 fpga user guide. *Product User Guide UG070*, 2008.
- [108] XILINX, INC. Command line tools user guide. *Product User Guide UG628*, 2009.
- [109] XILINX, INC. Virtex-4 fpga configuration user guide. *Product User Guide UG071*, 2009.
- [110] XILINX, INC. Memory interface solutions. *User Guide UG086*, 2010.
- [111] Z. A. YE, A. MOSHOVOS, S. HAUCK, and P. BANERJEE. Chimaera: a high-performance architecture with a tightly-coupled reconfigurable functional unit. In *ISCA*, pages 225–235, 2000.
- [112] A. S. ZEINEDDINI and K. GAJ. Secure partial reconfiguration of fpgas. In *Proceedings of the 2005 IEEE International Conference on Field-Programmable Technology, FPT*, pages 155–162, 2005.
- [113] Z. ZHOU, S. CHENG, and Q. LIU. Application of ddr controller for high-speed data acquisition board. In *ICICIC '06: Proceedings of the First International Conference on Innovative Computing, Information and Control*, pages 611–614, Washington, DC, USA, 2006. IEEE Computer Society.
- [114] R. ZIA-UR, J. DANIEL J., and W. GLENN A. A multiscale retinex for color rendition and dynamic range compression. Technical report, NASA, 1996.
- [115] P. ZIPF. Applying dynamic reconfiguration for fault tolerance in fine-grained logic arrays. *IEEE Trans. Very Large Scale Integr. Syst.*, 16(2):134–143, 2008.





# Eigene Veröffentlichungen

- [116] K. F. ACKERMANN, B. HOFFMANN, L. S. INDRUSIAK, and M. GLESNER. Enabling self-reconfiguration on a video processing platform. In *Third International Symposium on Industrial Embedded Systems, SIES*, pages 19–26, Juni 2008.
- [117] K. F. ACKERMANN, B. HOFFMANN, L. S. INDRUSIAK, and M. GLESNER. A lightweight sdram controller for self-reconfigurable video processing platforms. In *Workshop on Reconfigurable Communication-centric SoCs, ReCoSoC'08*, pages 63–70. Univ. Montpellier II, Juli 2008.
- [118] K. F. ACKERMANN, B. HOFFMANN, L. S. INDRUSIAK, and M. GLESNER. Providing memory management abstraction for self-reconfigurable video processing platforms. In *International Journal of Reconfigurable Computing*, pages 1–15, New York, NY, United States, Dezember 2009. Hindawi Publishing Corp.
- [119] K. F. ACKERMANN, L. S. INDRUSIAK, and M. GLESNER. System level design of a dynamically self-reconfigurable image processing system. In *Workshop on Reconfigurable Communication-centric SoCs, ReCoSoC'07*, pages 47–54. Univ. Montpellier II, Juni 2007.
- [120] K. F. ACKERMANN, T. LIU, and M. GLESNER. Extended distance transform approach for robust vehicle detection. In *Image and Signal Processing and Analysis, ISPA 2009*. IEEE, September 2009.
- [121] K. F. ACKERMANN, F. MAYER, L. S. INDRUSIAK, and M. GLESNER. Adaptable image processing system based on fpga modular multi kernel instantiations. In *Workshop on Reconfigurable Communication-centric SoCs, ReCoSoC'06*, pages 183–188. Univ. Montpellier II, Juli 2006.



# Betreute Arbeiten

- [122] A. ASHRAF. VHDL-Entwurf und Implementierung eines parametrisierbaren Algorithmus zur Echtzeit-Binärisierung von Videodaten. Diplomarbeit, Technische Universität Darmstadt, Juli 2005.
- [123] P. DICANDIA. Design-Automatisierung von dynamisch rekonfigurierbarer Hardware. Diplomarbeit, Technische Universität Darmstadt, November 2007.
- [124] T. LIU. Design and Software-Implementation of a Vehicle Classification Algorithm. Masterarbeit, Technische Universität Darmstadt, Februar 2009.
- [125] A. E. MAHJOUB. VHDL Hardware Entwurf und Implementierung dynamisch rekonfigurierbarer Module auf einer Virtex-4 FPGA Plattform. Diplomarbeit, Technische Universität Darmstadt, August 2006.
- [126] F. MAYER. VHDL-Hardware-Entwurf und Implementierung eines modularen Bildverarbeitungsalgorithmus für Lichtschnittverfahren. Diplomarbeit, Technische Universität Darmstadt, September 2004.
- [127] Q. QIANSHANG. Hardware Design and Optimization of a Control-unit for Dynamic Self-Reconfiguration of FPGAs. Masterarbeit, Technische Universität Darmstadt, April 2008.
- [128] C. SENEL. VHDL Hardware Design and Implementation of a Camera Link Framegrabber with Dynamic Self-reconfigurable Image Processing Cores. Masterarbeit, Technische Universität Darmstadt, Mai 2007.
- [129] S. ZINK. VHDL-Hardware-Entwurf und Implementierung modularer Algorithmen zur Farbrückgewinnung und Transformation eines CMOS Videosignals. Diplomarbeit, Technische Universität Darmstadt, Juli 2005.



# Lebenslauf

**Kurt Franz Ackermann**

---

## **Zur Person:**

Geburtstag                      1. Mai 1974  
Geburtsort                      Hatzfeld, Rumänien

## **Akademischer Werdegang:**

1980 - 1984                      Grundschule „Ludwig-Erk-Schule“ in Langen  
1985 - 1986                      Förderstufe „Adolf-Reichwein-Schule“ in Langen  
1987 - 1993                      Gymnasium „Dreieichschule“ in Langen  
Abschluss: Allgemeine Hochschulreife (*Abitur*)  
1993 - 1994                      Grundwehrdienst in Stadtallendorf  
1994 - 2001                      Student am Fachbereich Elektrotechnik und Informationstechnik, Technische Universität Darmstadt  
Abschluss: Diplom  
2001 - 2010                      Forschung und Entwicklung in der Bildverarbeitung bei VITRONIC Dr.-Ing. Stein Bildverarbeitungssysteme GmbH, Wiesbaden  
2004 - 2010                      Externer Doktorand am Institut für Mikroelektronische Schaltungen, Technische Universität Darmstadt

---

